## REMARKS

Reconsideration of the above-identified patent application in view of the amendments above and the remarks following is respectfully requested.

Claims 1, 3-9, 11, 12, 14-31, 33-39, 41, 42 and 49-59 are in this case. Claims 20-30 and 50-59 were withdrawn by the Examiner from consideration as drawn to a non-elected invention. Claim 16 has been rejected under § 112, second paragraph. Claims 1, 3-9, 11, 12, 14-19, 31, 33-39, 41, 42 and 44-49 have been rejected under § 103(a). Dependent claims 15 and 45 have been canceled. Independent claims 1 and 31 and dependent claims 3, 11, 12, 16, 19, 33, 41, 42, 46 and 49 have been amended. New dependent claims 64-66 have been added.

The claims before the Examiner are directed toward a network adapter, such as a host channel adapter (HCA), and a method of its use. The network adapter includes a host interface, an outgoing packet generator, an incoming packet processor, a network output port and a network input port. The host interface couples the network interface adapter to a host processor. The network output port transmits, via a network, outgoing request packets to remote responders and outgoing response packets to remote requesters. The network input port receives, via the network, incoming response packets from remote responders and incoming request packets from remote requesters. The incoming packet processor receives and processes both incoming response packets and incoming request packets such as incoming read request packets. The outgoing packet generator generates outgoing request packets as requested by the host processor and also generates outgoing response packets in response to incoming request packets.

The outgoing packet generator includes a gather engine that gathers, from a system memory accessible via the host interface, and via a common data flow path,

both write data for outgoing request packets and read data for outgoing response packets. Responsive to an incoming read request packet that specifies data to be read from a system memory accessible via the host interface, the incoming packet processor writes a response descriptor to a memory location, in a memory separate from the network adapter, indicating the data to be read from the system memory. The outgoing packet generator then reads the response descriptor from the memory location and, responsive thereto, generates the outgoing response packet.

**§ 112, Second Paragraph Rejections**

The Examiner has rejected claim 16 under § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Specifically, the Examiner has pointed out that the phrase "such" renders the claim indefinite because it is unclear whether the limitations following the phrase are part of the claimed invention.

Claim 16 now has been amended to clarify that all the recited response descriptors and all the recited outgoing response packets are part of the claimed invention. Claim 46, in which the phrase "such" also appears, has been amended similarly. In the specification, the response descriptors in the list of response descriptors are *e.g.* descriptors 110 of Figure 6. As described on page 26 lines 1-11, execution engines 92 and SDE 66 generate corresponding outgoing response packets.

**§ 103(a) Rejections – Pettey et al. '712 in view of Gasbarro et al. '004**

The Examiner has rejected claims 1, 3-9, 11, 12, 14-19, 31, 33-39, 41, 42 and 44-49 under § 103(a) as being obvious over Pettey et al., US Patent No. 6,594,712 (henceforth, "Pettey et al. '712") in view of Gasbarro et al., US Patent No. 6,948,004

(henceforth, "Gasbarro et al. '004"). The Examiner's rejection is respectfully traversed.

Claims 15 and 45 have been canceled, thereby rendering moot the Examiner's rejection of these claims.

Pettey et al. '712 teach an InfiniBand target channel adapter (TCA) **202** of an InfiniBand I/O unit **108**. The overall structure of TCA **202** is illustrated in Figure 3. TCA **202** includes a transaction switch **302**, a bus router **306**, IB MACs **308** and PCI bus interfaces **312** and **316**. Bus router **306** performs InfiniBand transport layer operations. IB MACs **308** are the interfaces between TCA **202** and an InfiniBand fabric **114**. PCI bus interfaces **312** and **316** are the interfaces between TCA **302** and a host of TCA **302**. Transaction switch **302** directs packets, datagrams and command messages between IB MACs **308**, bus router **306** and PCI bus interfaces **312** and **316**. (Transaction switch **302** also includes packet memory blocks **304** in support of direct DRMA operations, which is the invention of Pettey et al. '712). The packets that TCA **202** can handle include the packets illustrated in Figures 10-13: SEND packets **1000**, RDMA write packets **1100**, RDMA read request packets **1200** and RDMA read response packets **1300**. It follows that the correspondences between components of the present invention, as recited in claims 1 and 31, and components of prior art TCA **302** are as in the following table:

| Present invention | TCA **202** |
|---|---|
| Host interface | PCI bus interfaces **312** and **316** |
| Network output port | IB MAC **308** |
| Network input port | IB MAC **308** |
| Outgoing packet generator | Bus router **306** |
| Incoming packet processor | Bus router **306** |

Figure 14 of Pettey et al. '712 is a block diagram of the logical structure of bus router **306**. Bus router **306** includes work queue management logic **1412** for processing InfiniBand work queue requests, transmit packet process logic **1414** for creating outgoing packets, receive packet process logic **1416** for processing incoming packets and completion process logic for maintaining InfiniBand completion queues. Bus router **306** also includes a work queue memory **1402** in support of work queue management logic **1412**, a TxPP scratchpad memory in support of transmit packet process logic **1414** and a RxPP scratchpad memory in support of receive packet process logic **1416**.

Pettey et al. '714 are silent concerning whether bus router **306** includes the equivalent of a gather engine that gathers write data for outgoing write request packets and read data for outgoing read response packets via a common data flow path, as recited in independent claims 1 and 31. The Examiner has identified the interface adapter illustrated in Figure 7 of Gasbarro et al. '004 as including the functionality of the gather engine of the present invention. Applicant respectfully disagrees, and reserves the right to traverse this interpretation of Gasbarro et al. '004 later in the prosecution of the above-identified patent application. Instead, in a communication filed on October 19, 2006, Applicant distinguished the present invention from the prior art cited by the Examiner by amending claims 1 and 31 to include the limitations of claims 10 and 40, respectively.

The following arguments in defense of these amendments were presented in the communication filed on October 19, 2006:

> ...the limitation of claims 10 and 40 that distinguishes the present invention from the prior art cited by the Examiner is that the incoming packet processor prepares a read response work item, in a memory location, that is read by the outgoing packet generator to tell the outgoing packet generator how to generate the corresponding read

response packet. In the embodiment of the present invention that is described in the specification, this read response work item is the "quasi-WQE" that is prepared by TCU 52 in RDB 40 in response to the receipt of a RDMA read request from a remote requester, as described in the specification on page 19 line 27 through page 20 line 7. As described in the specification on page 20 lines 8-19, execution unit 60 and SDE 66 treat quasi-WQEs the same as WQEs received from host 24. It is this emulation of the standard InfiniBand WQE mechanism by the network adapter (HCA 22 in the embodiment described in the specification) for remote requester RDMA read requests, independently of the host of the network interface adapter, that enables the present invention to use the same gather engine (SDE 66 in the embodiment described in the specification) to gather data for both requester and responder outgoing packets via the same data flow path.

By contrast, the prior art cited by the Examiner uses the standard InfiniBand WQE mechanism, in which WQEs are generated by the host, to specify data to be sent to a remote requester in response to a RDMA read request. For example, Pettey et al. '714 illustrate in Figure 18a their direct RDMA write in response to the receipt of a SEND packet as illustrated in Figure 15. The location in local memory **218** of the data to be sent to the requester is specified in step **1808** of Figure 18 by a WQE generated by host CPU **208**, as described in column 17 lines 21-22:

> Next the CPU **208** creates a DRDMA Write WQE **800** of FIG. 8 and submits it to the TCA **202**, in step **1808**.

There is neither a hint nor a suggestion in the prior art cited by the Examiner of a channel adapter such as TCA **202** of Pettey et al. '714 generating its own WQE equivalents, independently of its host, in response to packets received from remote requesters.

In response to these arguments, the Examiner now has asserted that the limitation, that the read response work item includes information that tells the outgoing packet generator how to generate the corresponding read response packet, is lacking from the claims. Therefore, the limitations of claims 10 and 40 have been replaced, in claims 1 and 31, by the limitations of claims 15 and 45 that recite this aspect of the present invention more explicitly. Specifically, claim 15 recites the limitation that the response descriptor indicates the data to be read from the system memory responsive to the incoming read request packet, and claim 45 recites the

limitation that the incoming read request packet specifies data to be read from a system memory associated with the host processor. Correspondingly, claims 15 and 45 have been canceled, claims 16 and 19 have been amended to depend directly from claim 1, and claims 46 and 49 have been amended to depend directly from claim 31.

In rejecting claims 15 and 45, the Examiner has cited Pettey et al. '712 Figure 18a step **1822** as teaching the writing of a response descriptor to a memory location indicating data to be read from system memory responsive to a read request packet. Applicant respectfully disagrees. The memory location that is the target of step **1822** is a PMB **304** in transaction switch **302** that, as illustrated in Figure 3, is part of TCA **202**. By contrast, the memory locations in which the response descriptor of the present invention is written are outside the network interface adapter. As stated in the specification on page 19 line 31 through page 20 line 4,

> Preferably, for the sake of efficiency, some or all of the RDB is held in an off-chip memory 67 coupled to the HCA, which is accessed using a double-data-rate memory controller (DMU) 65, rather than via the system bus. Additionally or alternatively, the RDB may be held in system memory 38, as shown in Fig. 1.

To clarify this matter, claim 1 has been amended further to state that the first memory location is in a memory separate from the network interface adapter, and claim 31 has been amended to state that the first memory location is in system memory. In addition, new claim 64 has been added to limit the memory separate from the network interface adapter to be the system memory.

These amendments of claims 1 and 31 also required the following amendments of the claims:

Claim 3 recites a "first" memory location where the host processor writes a request descriptor indicative of the write data and a "second" memory location where the incoming packet processor writes a response descriptor indicative of the read data.

With the writing of the response descriptor by the incoming packet processor now recited in claim 1, the latter limitation has been deleted from claim 3, the memory location where the incoming packet processor writes the response descriptor now is called the "first" memory location, and the memory location where the host processor writes request descriptor now is called the "second" memory location.

The response descriptor was called a "read response work item" in claims 10-12 and in claims 40-42. Therefore, "read response work item" has been changed to "response descriptor" in claims 11, 12, 41 and 42.

Claim 33 recites a "first" memory location to which a request descriptor indicative of the write data is generated and a "second" memory location to which a response descriptor indicative of the read data is written. With the writing of the response descriptor now recited in claim 31, the latter limitation has been deleted from claim 33, the memory location where the response descriptor is written now is called the "first" memory location, and the memory location to which the request descriptor is generated is called the "second" memory location in claims 33 and 49.

The Examiner also has stated:

> If applicant's intention is directed towards the subject matter "Quasi-WQE", applicant is advised to distinctly and clearly disclose the subject matter in the independent claims.

Based on page 25 lines 10-11 of the specification,

> These descriptors are referred to herein as "quasi-WQEs."

Applicant believes that it is not necessary to call the response descriptors "quasi-WQEs" in the claims. Nevertheless, in order to expedite the prosecution, Applicant has added new claims 65 and 66 that state that the response descriptors of claims 1 and 31 are quasi-WQEs.

The Examiner also has asserted that

> ...the claim fails to disclose, teach or suggest "same gather engine to gather data for both the requestor and responder outgoing packet via the same data flow path".

Applicant respectfully disagrees. Claim 1 recites

> ...a gather engine, which is coupled to gather both the write data and the read data from the system memory, for inclusion in the respective outgoing packets via a common data flow path... (emphasis added)

Claim 31 recites

> ...generating the outgoing write request packet and generating the outgoing read response packet comprise generating the packets using a gather engine...which is coupled to gather both the write data and the read data from the system memory for inclusion in the respective outgoing packets via a common data flow path... (emphasis added)

Finally, the Examiner has cited Pettey et al. '712 column 11 lines 21-29:

> When the CPU **208** of FIG. **2** desires to send the host **102** a message, it submits a work request **722** to the TCA **202** Send Queue **714**. The TCA **202** creates a Work Queue Entry (WQE) and places the WQE on the Send Queue **714**. Among the WQE types are RDMA Write WQE **762**, RDMA Read WQE **763**, DRDMA Write WQE **764**, DRDMA Read WQE **765**,and SEND WQE **766**.

as teaching generation of WQE equivalents by TCA **202**, independent of host **102**. Applicant respectfully points out that what this citation from Pettey et al. '712 really teaches is generation of WQEs by CPU **208**, not by TCA **202**. As illustrated in Figure 2 of Pettey et al. '712, CPU **208** and TCA **202** are separate components of an IB I/O unit **108**. Within IB I/O unit **108**, CPU **208** is the host of TCA **202**. The generation of WQEs by CPU **208** is the prior art generation of WQEs by a client process running on a host, as described in the specification on page 2 lines 8-15.

For completeness, Applicant also addresses a question that the Examiner asked concerning a statement in the communication filed on October 19, 2006, that Pettey et al. '714 are silent concerning the internal architecture of the various logics of bus router **306**. The Examiner stated that it is not clear what other "various logics" Applicant was referring to. The logics of bus router **306** are Work Queue

Management logic **1412**, Transmit Packet Process logic **1414**, Receive Packet Process logic **1416** and Completion Process logic **1418**, as illustrated in Figure 14 and as described in column 14 line 10 through column 15 line 4.

With independent claims 1 and 31 allowable in their present form it follows that claims 3-9, 11, 12, 14, 16-19, 33-39, 41, 42, 44 and 46-49 that depend therefrom also are allowable.

In view of the above amendments and remarks it is respectfully submitted that independent claims 1 and 31, and hence dependent claims 3-9, 11, 12, 14, 16-19, 33-39, 41, 42, 44 and 46-49 are in condition for allowance. Prompt notice of allowance is respectfully and earnestly solicited.

Respectfully submitted,

Mark M. Friedman
Attorney for Applicant
Registration No. 33,883

Date: April 30, 2007